Electronic Version
Stylesheet Version v1.1.1

Claims

[c1] What is claimed is:

1. A very long instruction word (VLIW) architecture comprising:

a VLIW input port for sequentially inputting a plurality of VLIWs, each VLIW comprising a plurality of instructions;

a decoder for decoding the instructions of the VLIWs;

at least a register for storing data;

a plurality of data buses for sending data;

a plurality of arithmetic logic units (ALUs) for executing the instructions of the VLIWs; and

a plurality of multiplexers, each output port of the multiplexers being connected to an input port of one of the corresponding ALUs, and each input port of the multiplexers being connected to the register and output ports of the ALUs via the data buses;

wherein each of the multiplexers selects two outputs from outputs of the register and the ALUs to send to the corresponding ALU so that the corresponding ALU executes one of the instructions to operate the two selected outputs.

[c2] 2. The VLIW architecture of claim 1 wherein each multi-

plexer is connected to the decoder, and the multiplexer selects the two outputs from outputs of the register and the ALUs according to the instructions decoded by the decoder.

[c3] 3. The VLIW architecture of claim 1 wherein each multiplexer periodically selects the two outputs from outputs of the register and the ALUs, and sends the selected two outputs to the corresponding ALU so that the ALU periodically executes the instructions to operate the two selected outputs.

[c4] 4. The VLIW architecture of claim 1 wherein each instruction comprises a scheduling flag, and the decoder decides the order that the ALUs execute the instructions according to the scheduling flags of the instructions.

[c5] 5. The VLIW architecture of claim 1 further comprising a VLIW register connected to the VLIW input port and the decoder for storing the VLIWs input from the VLIW input port.

[c6] 6. The VLIW architecture of claim 1 wherein the output port of each multiplexer connects to the register, and each multiplexer selects an output of the ALUs to store in the register.

[c7] 7. A very long instruction word (VLIW) architecture com-

prising:

a VLIW input port for sequentially inputting a plurality of VLIWs, each VLIW comprising a plurality of instructions;

a decoder for decoding the instructions of the VLIWs;

a register file for storing data, the register file comprising a plurality of registers;

a plurality of data buses for transferring data;

a plurality of arithmetic logic units (ALUs) for executing the instructions of the VLIWs; and

a plurality of multiplexers, each output port of the multiplexers being connected to an input port of one of the corresponding ALUs, and each input port of the multiplexers being connected to the register and output ports of the ALUs via the data buses;

wherein each of the multiplexers selects two outputs from outputs of the register and the ALUs to send to the corresponding ALU so that the corresponding ALU executes one of the instructions to operate the two selected outputs.

[c8] 8. The VLIW architecture of claim 7 wherein each multiplexer is connected to the decoder, and selects the two outputs from outputs of the register and the ALUs according to the instructions decoded by the decoder.

[c9] 9. The VLIW architecture of claim 7 wherein each multiplexer periodically selects the two outputs from outputs

of the register and the ALUs, and sends the selected two outputs to the corresponding ALU so that the ALU periodically executes the instructions to operate the two selected outputs.

[c10] 10. The VLIW architecture of claim 7 wherein each instruction comprises a scheduling flag, and the decoder decides the order that the ALUs execute the instructions according to the scheduling flags of the instructions.

[c11] 11. The VLIW architecture of claim 7 further comprising a VLIW register connected to the VLIW input port and the decoder for storing the VLIWs input from the VLIW input port.

[c12] 12. The VLIW architecture of claim 7 wherein the output port of each multiplexer connects to the registers, and each multiplexer selects an output of the ALUs to store in one of the registers.